PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

CN

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06K 9/38 | A1

(11) International Publication Number: WO 93/04442

(43) International Publication Date: 4 March 1993 (04.03.93)

(21) International Application Number: PCT/US92/07054

(22) International Filing Date: 24 August 1992 (24.08.92)

(30) Priority data:
07/749,190 23 August 1991 (23.08.91) US

(71) Applicant: UNITED PARCEL SERVICE OF AMERICA, INC. [US/US]; 400 Perimeter Center, Terraces North, Atlanta, GA 30346 (US).

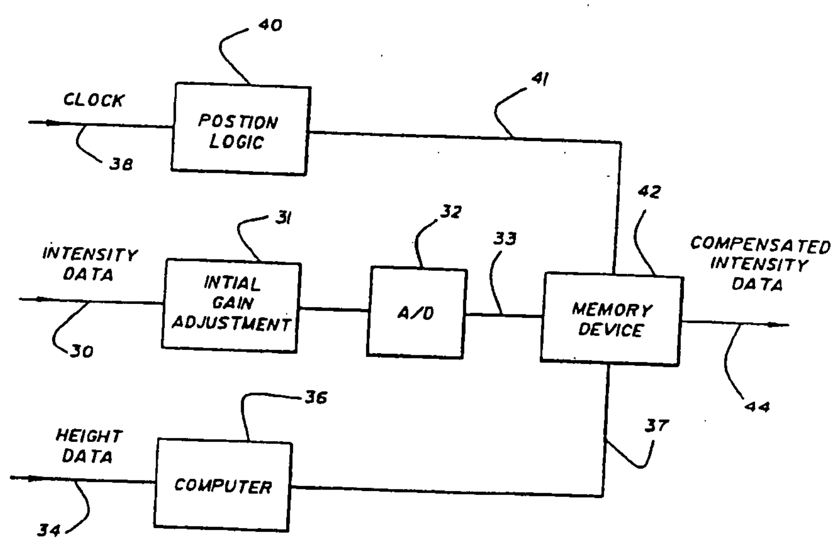
(72) Inventor: SHAH, Govind ; 11 Amherst Way, Princeton Junction, NJ 08550 (US).

(74) Agents: DROBILE, James, A. et al.; 1600 Market Street, Suite 3600, Philadelphia, PA 19103 (US).

(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR COMPENSATION FOR NON-UNIFORM ILLUMINATION



(57) Abstract

A method for illumination compensation in a system for processing of signals representing reflected light intensities from a surface includes the steps of receiving reflected light intensity information in the form of more than one intensity value (33) and receiving, associated with each one of the reflected light intensity values, a value of at least one variable (37, 41). The method further includes remapping (42) each of the reflected light intensity values to one or more than one compensated reflected light intensity values, including remapping any of the reflected light intensity values equal to or less than a preselected threshold minimum reflected light intensity value associated with the value of each of the variables for that reflected light intensity value, to a minimum level output, and remapping any of the reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value, associated with the value of each of the variables for that reflected light intensity value, to a maximum level output.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/07054

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(5) :G06K 9/38 US CL :382/50 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 382/50, 53		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) none		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 4,314,281 (WIGGINS) 02 February 1982 (02.02.82). See col. 27, line 12 through col. 28, line 46.	1-9
Y	US, A, 4,430,748 (TUHRO ET AL.) 07 February 1984 (07.02.84). See col. 2, lines 58-68.	1-4
Y	US, A, J. SIMON AND A. ROSENFELD, "DIGITAL IMAGE PROCESSING AND ANALLYSIS", Published 1977, Noordhoff International Publishing, pages 122-125.	1-9

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 02 NOVEMBER 1992	Date of mailing of the international search report 10 DEC 1992
Name and mailing address of the ISA/ US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE	Authorized officer [illegible] LEO H. BOUDREAU [illegible]-HO INTE[illegible]IONAL DIVISION Telephone No. (703) 305-4762

Form PCT/ISA/210 (second sheet)(July 1992)★

METHOD AND APPARATUS FOR COMPENSATION FOR NON-UNIFORM ILLUMINATION

Background of the Invention.

This invention relates to compensation for non-uniformity in illumination of a surface in a system that takes information in optical form from the surface and converts such information to electronic form and in particular to compensating for variations in illumination in systems for reading labels on articles, such as packages or letters, being moved at high speed along a conveyor belt past a video camera.

In the shipping of parcels and packages, it is necessary, at various points in the process, to obtain various information that may be encoded on a label. Such information includes address information, shipper information, and the like. Encoded labels also have application in other fields, such as manufacturing. Uses for information encoded on a label may include, by way of example, sorting packages further along the conveyor belt, determining the quantity of items going to a particular location, and recording the movement of packages so as to provide updates in response to customer inquiries. The rapid obtaining of information from a label may be facilitated by an automated system for converting optically encoded information on the labels to electrically encoded information.

In modern terminals for sorting of packages, packages are loaded on a conveyor belt moving at speeds of up to 100 inches per second and moved to various points for sorting. Such conveyor belts are ordinarily three to four feet wide. A common width is 42 inches. Packages may be loaded onto such conveyor belts anywhere across their widths, and in any orientation. Packages of differing sizes may be shipped along a single conveyor belt. As a result when a package passes through a zone for reading of label information, the label may be at any point across the width of the conveyor belt and at any height within a range of heights above the surface of the conveyor belt.

Objects of the Invention.

Accordingly, it is an object of this invention to provide a method and an apparatus for correcting signals representative of information stored on a surface in optical form detected under conditions of non-uniformity of illumination of the surface.

It is a further object of this invention to provide such a method and apparatus for use in a system in which labels are placed on packages of varying heights.

It is a further object of this invention to provide a method and apparatus for correcting signals representing label information in which labels are placed on packages that are moving on a conveyor belt.

It is a particular object of this invention to provide such a method and apparatus for use where the information is stored on the surface in white areas and black areas.

Additional objects and advantages of the invention will become evident from the detailed description of a preferred embodiment below.

Summary of the Invention.

A method for illumination compensation for use in a system for reading surfaces having at least black and white areas, includes the steps of storing reference reflected light intensity information corresponding to at least black and white areas in the form of the plurality of intensity values, each one of the intensity values having an address defined by a value of at least one variable, calculating from the reference reflected light intensity information more than one black threshold value and more than one white threshold value each of which threshold values is associated with a selected value of each of the variables. The method further includes the steps of receiving opera-tional reflected light intensity information in the form of more than one operational reflected light intensity value, each of the operational reflected light intensity values having an address defined by a value of each of the variables, remapping each of the operational reflected light intensity values to one of at

a preselected threshold minimum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a minimum level output and memory means for remapping any of the reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a maximum level output.

Brief Description of the Drawings.

Figure 1 is a simplified perspective view of a system for use with a method and apparatus according to the invention.

Figure 2 is a block diagram of a circuit according to the invention.

Figure 3 is a partial view of an calibration bar for use in connection with the invention.

Figure 4 is a flow chart of the overall process of calibration in connection with a method and apparatus according to the invention.

Figure 5 is a graph showing reference reflected light intensity data versus horizontal position before simplification.

Figure 6 is a flow chart showing the steps of reflected light intensity data simplification.

Figure 7 is a graph showing reference reflected light intensity data versus horizontal position after data simplification.

Figure 8 is a graph showing output reflected light intensity values versus input reflected light intensity values as remapped according to the invention.

Detailed Description of a Preferred Embodiment.

With reference to Figure 1, there is shown an example of an apparatus for reading surfaces having information, such as reading label information on articles being moved on a conveyor belt. Such an apparatus may be used in connection with an illumination compensation apparatus of the invention. There is shown a section of a conveyor belt 60. Conveyor belt 60 is moving in the direction of arrow 61. Box 62 is on the conveyor belt 60.

to discrete pixels. The reflected light intensity information is in the form of a series of discrete values or reflected light intensity values. Each discrete value corresponds to a particular position along a scan line.

A second type of information is information relating to the values of the variables, or variable value information, upon which the illumination depends. In the disclosed embodiment of the invention, there are two variables upon which the illumination depends. One such variable is height. Height information is transmitted along line 34 to a computer 36.

The second such variable is horizontal position. The horizontal position of any item of reflected light intensity is preferably derived from a clock signal associated with the reflected light intensity information transmitted by the light-detecting device. A clock signal from the light-detecting device is transmitted along line 38 to position logic unit 40. The light-detecting device also transmits a signal to the position logic unit 40 representing the beginning of a new scan line. From the clock signal and the new scan line signal, position logic unit 40 provides a position signal. The unit may be an Altera 5032. The providing of such a signal is conventional in the art. The position signal is sent simultaneously with each intensity value. The position signal counts from zero on each scan line across the illumination zone. The position signal value is changed a certain number of times in each scan line depending on detail of compensation in the apparatus. This is explained below in more detail where the look-up tables are discussed.

Computer 36 is shown in Figure 2. As noted above, an input of computer 36 receives height information. Computer 36 has the function of processing height information in the system of the invention. One output of computer 36 is coupled to an input of initial gain unit 31. This output is a gain signal. In a preferred embodiment, this gain signal is an 8-bit digital signal. This gain signal is used as an adjustment for gain variation relating only to height. It has been found that reflected

of two variables define the address of each reflected light intensity value.

During calibration of an apparatus according to the invention, the reflected light intensity value information represents light reflected from a surface having a known, or reference, optical property. Such reflected light intensity value information is referred to as reference reflected light intensity value information.

During operation of an apparatus according to the invention, the reflected light intensity value information represents light reflected from a surface of unknown optical properties. Such reflected light intensity value information is referred to as operational reflected light intensity value information.

The initial gain adjustment apparatus will now be explained in greater detail. An 8-bit digital height signal is provided from the computer. The signal is input to a multiplier such as a CA 3338. The output of this logic unit is a gain-adjusted signal containing reflected light intensity value information. This is an analog signal. The amplitude of the output signal is determined by the height signal provided by the computer. In the preferred embodiment, a chip is provided that has 256 possible gain adjustments.

The height signal is generated by the computer depending on the height of the package. In one embodiment, a height sensor is employed, over a range of 28 inches, with a sensitivity of one-half inch. Thus, in that embodiment 56 different height readings are possible. In another embodiment, the height sensor is used over a range of 36 inches, thus making 72 different height readings possible. The computer has a look-up table stored in memory of gain signal values corresponding to each possible height value. The entries in the look-up table are obtained during calibration of the apparatus. The calibration of the apparatus is explained in detail below.

The output of the gain adjustment unit is coupled to an analog-to-digital conversion unit. The unit

changes 16 times in every sweep of the CCD across the width of the belt. The position signal is provided to at least two memory devices corresponding to left and right sides of the illumination zone. In a preferred embodiment, there are four channels provided. Two channels receive reflected light intensity value information corresponding to the left half of the illumination zone. Two other channels receive reflected light intensity value information corresponding to the right side of the illumination zone. Thus the position value has 16 possible values, while 32 horizontal segments are provided. The same position signal corresponds to two position variable values, depending on which channel received the signal. The position value signal is thus a four-bit signal for each half of the illumination zone. Each time the position value changes, a new table is designated on each half of the illumination zone.

In the preferred embodiment, a CCD imager is employed having 4,096 pixels in a scan line. Thus each table remaps 128 reflected light intensity values.

The operation of the look-up tables in the memory devices is best understood with reference to Figure 8. Figure 8 is a graph of output versus input for three horizontal segments at a single height range. Figure 8 thus represents remapping of input to output for three of the 32 tables in a single height range.

Line 802, which is a staircase curve, represents the remapping of the input values to the output values by one table. Line 802 corresponds to segment 2, which remaps values of pixels 128-255. All input digital intensity values less than a black level threshold value are remapped to a black output level. In the illustration of Figure 8, the black output level is zero. The black level threshold value for line 802 is approximately 102. All input values above a white threshold are remapped to a white output level. For the segments illustrated in Figure 8, the white output level is the maximum output level of the memory device. The white threshold for line 802 is approximately 192.

In such event, a reference reflected light intensity value would be obtained using a calibration bar having areas with whatever optical property represented the minimum reflected light intensity value. This reference reflected light intensity value would be designated a threshold minimum reflected light intensity value. The memory means would store a look-up table designed to remap any operational reflected light intensity value equal to or less than the threshold minimum reflected light intensity value obtained from the reference to a minimum output level.

Similarly, it is possible to use the invention in a context in which there is a maximum reflective light intensity value that corresponds to an optical property of a surface other than white. In such a system, there would be a surface used for calibration that had the optical property corresponding to the maximum reflected light intensity value used for determining a reference maximum reflective light intensity value. This reference reflected light intensity value would be designated a threshold maximum reflected light intensity value. The memory device according to the invention would be programmed with a look-up table that remapped all operational reflected light intensity values equal to or greater than the threshold maximum reflected light intensity value to a maximum output level.

It will be seen that in the general case, any operational reflected light intensity value falling between the minimum reflected light intensity value threshold and the maximum light intensity value threshold would be remapped to an output value between a minimum output value and a maximum output value. For example, the values of operational reflected light intensity value falling between the minimum reflected light intensity value threshold and the maximum reflected light intensity value threshold can be remapped, to an output value between a minimum output value and a maximum output value. However, the intermediate values could be remapped according to any desired function.

storage in a memory device. This is shown in box 516 in Figure 4. The last step is the step, shown in box 518 of Figure 4, of storing the tables in the memory devices.

In making these calibration adjustments, it is desirable to use a calibration bar 100 as shown in Figure 3. Alternating white strips 102 and black strips 104 are provided on the calibration bar 100 perpendicular to the length of the bar. For example, each strip may be 0.027 inches in width. The calibration bar 100 is sufficiently long that it can rest on pegs 82 of Figure 1, and extends across the entire belt. It will be seen that for obtaining the reference reflected light intensity values, a surface or surfaces must be provided having both the optical property corresponding to the minimum reflected light intensity value and the optical property corresponding to the maximum reflected light intensity value.

In the first step, shown as box 502, the attenuation gain is adjusted. The adjustments are made using an oscilloscope to monitor the levels. The goal in adjusting the attenuation gain is to assure that the black level is not below zero, and that the white levels are not saturated. The calibration bar is placed at various selected heights. The step of adjusting the gain is repeated at each selected height. The gain value for each selected height is stored in the computer for later processing. In a preferred embodiment, the selected heights are 1 inch, 4 inches, 8 inches, 12 inches, 16 inches, 20 inches, 24 inches and 28 inches above the conveyor belt. In a system for reading surfaces having heights up to 36 inches, additional readings would be taken.

The gain values set in the calibration process represent eight height points. In fact, in a preferred embodiment of the invention, there is provided height sensing means having the capability of detecting half-inch height intervals. The points intermediate the eight selected heights are obtained by standard curve-fitting techniques. In a preferred embodiment, the identical curve-fitting technique and software that is used as described below in the step of simplification of the com-

tially in the form of two lines. With reference to Figure 5, the horizontal axis represents horizontal position across the illumination zone. The vertical axis represents reflected light intensity. In Figure 5, thirty-two data points, or pixels, have been chosen for display in the figure. The intensity values shown have been measured by placing an input of an oscilloscope on an intensity output line from the light-detecting device. The intensity value numbers are arbitrary.

The line designated by 602 in Figure 5 shows the white levels. As may be seen, there is substantial variation in the white levels. In particular, the white levels drop on the edges of the illumination zone. The line designated by 604 represents the black levels. As may be seen, there is also considerable variation in the black levels. A profile similar to Figure 5 is obtained for each height interval at which the calibration bar is placed. The set of raw profiles are stored for further processing. Each raw profile is stored in a separate file in the computer.

After collection and storing of the reference intensity values, there is an optional step of checking the reference intensity value information data for evidence of hardware failure. A hardware failure in the system would make reading of labels with a reasonable degree of accuracy impossible. Such hardware failures include a failure of a portion of the illumination apparatus, such as a bulb, or of a portion of the light sensing means, such as a failure of a CCD array.

Checking the profile data for evidence of hardware failures is accomplished by computing a variance and standard deviation. The equation for the variance, var $(X_{1..n})$, is the following:

$$\mathrm{var}(x_{1...n}) = \frac{1}{N-1} \times \sum_{j=1}^{N} \left(x_j - \left(\frac{1}{N} \times \sum_{j=1}^{N} x_j\right)\right)^2$$

where X is the intensity value and N is the number of pixels in a profile. The standard deviation $\sigma(x_{1..n})$ is defined by the equation of $\sigma(x_{1..n}) = \sqrt{\mathrm{var}(x_{1..n})}$

tion zone. The equation for finding the fitted curve from the raw profiles is the following:

$$\begin{bmatrix} m & \sum x & \sum x^2 \\ \sum x & \sum x^2 & \sum x^3 \\ \sum x^2 & \sum x^3 & \sum x^4 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum x \cdot y \\ \sum x^2 \cdot y \end{bmatrix}$$

The source code for this is shown in CURVE-FIT C, which is included herewith in the appended software listings.

Referring to Figure 6, there is shown a flow chart depicting the steps of the process of data simplification employing a preferred curve-fitting technique. In the step illustrated by the box labeled 702 on Figure 6, there is shown the step of allocating arrays. The allocation of arrays is a conventional preliminary step to performing vector and matrix mathematics. The next step is the step of inserting the X and Y values in the matrices and arrays. The X values are the pixel numbers, which correspond to horizontal position in the illumination zone. The Y values are reflected light intensity values. This step is shown at box 704 in Figure 6. The next step, shown in box 706 in Figure 6, is the step of computing the X summations. The X summations are elements of the left-hand matrix in the above equation for the curve-fit. This step requires computation of summations of an array of X-values raised to various powers. A preferred software subroutine for this computation is shown under the file name "sum_a" in the appended source code listings. The next step in the process is the step, shown in box 708 in Figure 6, of computing the XY summations. This step requires taking two arrays of data, the array of X-values and the array of Y-values, and calculating a summation. The values in the array of Y-values are raised to increasing powers. A software subroutine for this step is shown under the file name "sum_ab" in the appended source code listings. The next step, shown at box 710, is transferring the values obtained from the step of computing the X summations to a matrix. This is the matrix shown on the left hand side of the above equation for the second-degree

After the step of processing the raw profile data into curve-fitted profile data, there are four sets of profile data. Each set of profile data represents a profile corresponding to one of four ranges of height values in the illumination zone. Each set of profile data is next broken into 32 segments as shown by box 512 in Figure 4. The 32 segments represent the 32 horizontal segments of the illumination zone. Each of the 32 segments corresponds to 128 pixels across the illumination zone.

The simplified reference reflected light intensity values, or simplified profile data, as shown by box 514 in Figure 4, is then mapped to an inverse curve. In obtaining the inverse, each line is first rescaled by conventional techniques.

The values contained in the inverse curve are used in programming the illumination compensation memory devices. The values must be placed in a format that can be accepted by the memory devices. In a preferred embodiment, an Intel MCS-86 Hex file is created for each of the PROM's. The data stored is in the form of a table having 256 elements.

The data is then converted to a form which is compatible with the various channels into which the data is divided by the CCD units. For example, in a preferred embodiment, there are four channels. These four channels represent respectively odd and even pixels on the left and right sides of the belt. Accordingly, the data generated is divided into appropriate segments for each channel. Once the data is divided in this manner, the data is stored in computer memory in four parts. The data is transferred from the computer to the memory device. In a preferred embodiment, the memory device is an ultra-violet light erasable programmable read-only memory (PROM). A Cypress 7C254 may be used, for example. In order to transfer data from the computer to such a device, a PROM burner is provided. The input of the PROM burner is coupled to a port of the computer so that data may be transferred from the computer to the PROM burner. In

```
/**********************************************************************
* File Name: moment.c
*
* File Function: Routine to compute the average, standard deviation
*                and varaiance of the data passed.
*
*
* Notes: Function was taken from Numerical Recipe in 'C' see *
    chapter 13 pg 475
*        this function is a subset of the moment function.
*
* History: 1988 Created
*
**********************************************************************/

#include <math.h>

void moment(data,m,n,index,ave,sdev,svar)
int m,n,index;
float **data,*ave,*sdev,*svar;
{
   int j;
   float s;
   void nrerror();


   if (n <= 1)
    nrerror("n must be at least 2 in MOMENT");

   s=0.0;
   for (j = 1; j <= n;j++)
    s += data[index][j];

   *ave=s/n;

   *svar = 0.0;

   for(j = 1; j <= n; j++)
     *svar += fabs(s = data[index][j] - (*ave));


   *svar /= n;
   *sdev=sqrt(*svar);

}
```

```
curvefit(data,type,index,b)
 float **data,*b;
 int index;
 char type;
{
  float *x,*y,**a,*xsum,*xysum,*col,**ai,d,power;
  int i,j,k,m,*indx;
  FILE *fp;

  x = vector(1,VECMAX);              /* allocate array space */
  y = vector(1,VECMAX);
  xsum = vector(1,VECMAX);
  xysum = vector(1,VECMAX);
  a = matrix(1,ARRAMAX,1,ARRAMAX);
  ai = matrix(1,ARRAMAX,1,ARRAMAX);
  col = vector(1,VECMAX);
  indx = ivector(1,VECMAX);

  if(type == 'P')                 /* Dealing with profile gains */
   m = VECMAX;
  else
   m = VECMAX/4;                  /* Dealing with SA  gains */


  if(type == 'P')                 /* Profile Gains */
   {
     for(i = 1; i <= m; i++)      /* Get x values */
      x[i] = i;
   }
  else                            /* Sa gains */
   {
     for(i = 1, j = 1; i <= m; i++, j += 4)
      x[i] = j;                  /* Note Sa gains are increments 4 */
   }

  for(i = 1; i <= m; i++)         /* Get y values */
   y[i] = data[index][i];

  power = 1.0;                    /* Compute the x summations */
  for(i = 1; i <= SUMAX; i++)
   xsum[i] = sum_a(x,m,power++);


  power = 1.0;                    /* Compute the xy summations */
  xysum[1] = sum_a(y,m,power);

  for(i = 2; i < SUMAX; i++)
   xysum[i] = sum_ab(x,y,m,power++);
```

```
/*********************************************************************
* File Name: sum_a
*
* File Function: To take an array of data and find a summation *
*               raised to a variable power
*
* Notes:
*
* History: JSK 4-June-91 Created
*
*********************************************************************/

float sum_a(a,limit,power)
 float *a,power;
 int limit;
{
 int i;
 float sum = 0.0;

 for(i = 1; i <= limit; i++)
  sum += pow(a[i],power);

 return sum;
}

/*********************************************************************
* File Name: sum_ab
*
* File Function: To take two arrays of data and find a summation
*                  raised to variable power
*
* Notes:
*
* History: JSK 4-June-91 Created
*
*********************************************************************/

float sum_ab(a,b,limit,power)
 float *a,*b,power;
 int limit;
{
 int i;
 float sum = 0.0;

 for(i = 1; i <= limit; i++)
  sum += (pow(a[i],power) * b[i]);

 return sum;
}
```

```
/*********************************************************************
* File Name: ludcmp.c
*
* File Function: Given an nxn matrix routine replaces it by the LU
*                  decomposition  of a row wise permutation of *
            itself.
*
* Notes: Function was taken from Numerical Recipe in 'C' see *
    chapter 2 pg 43
* History: 1988 Created
*
*********************************************************************/

#include <math.h>

#define TINY 1.0e-20;

void ludcmp(a,n,indx,d)
int n,*indx;
float **a,*d;
{
     int i,imax,j,k;
     float big,dum,sum,temp;
     float *vv,*vector();
     void nrerror(),free_vector();

     vv=vector(1,n);
     *d=1.0;
     for (i=1;i<=n;i++) {
          big=0.0;
          for (j=1;j<=n;j++)
               if ((temp=fabs(a[i][j])) > big) big=temp;
          if  (big  ==  0.0)  nrerror("Singular  matrix  in  routine
LUDCMP");            vv[i]=1.0/big;
     }
     for (j=1;j<=n;j++) {
          for (i=1;i<j;i++) {
               sum=a[i][j];
               for (k=1;k<i;k++) sum -= a[i][k]*a[k][j];
               a[i][j]=sum;
          }
          big=0.0;
          for (i=j;i<=n;i++) {
               sum=a[i][j];
               for (k=1;k<j;k++)
                    sum -= a[i][k]*a[k][j];
               a[i][j]=sum;
               if ( (dum=vv[i]*fabs(sum)) >= big) {
                    big=dum;
                    imax=i;
               }
          }
```

```
/*
 *   File: genprom.c
 *   Auth: Robert K Pekarek
 *   Date: 03 Jan 1990
 *
 *   Copyright (c) 1990 United Parcel Service.
 *
 *   Purpose: Generate prom object files for SA correction tables.
 *
 *   Edit: Jim S Kunicki (JSK) added #define min marco to compile
 *         under Unix
 *         Changed progam to be a callable function
 */

#define HALVES 2
#define FIRST 0
#define SECOND 1
#define HZ 4
#define PZ 16
#define DATABITS 256
#define SIZE 16
#define VERSION 1
#define REVISION 0
#define min(a,b) (((a) < (b)) ? (a) : (b))


#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#ifdef MSDOS
#include <process.h>
#endif

/**********************************************************************
extern void main(int argc, char **argv);
static FILE *GetInputFile(int argCount, char **stringVector);
static void GetFSOpsRND(FILE *fp, int *full, float *preScale, int
*quantize,
          int *inputScale);
static void BuildRawOutOffset(FILE *fp, float *off1, float *raw1,
                                                   float    *off2,
float *raw2);
static void CalcProm(float *off1, float *raw1,
                         int full, float preScale, int quantize,
int inputScale,
                         unsigned char *result, float *scale);
static void WriteIntelHex(unsigned char *result, int aLen, char
*outFileName);
static void WriteText(unsigned char *result, char *outFileName);
static  void  WriteScaleFactors(float  *scale,  int  aLen,  char
*outFileName);
**********************************************************************/
```

```
     /* Write prom data to files */
     WriteIntelHex(&results[FIRST][0][0][0],     PZ*HZ*DATABITS,
"AB.PRM");
     WriteIntelHex(&results[SECOND][0][0][0],    PZ*HZ*DATABITS,
"CD.PRM");

     /* Write prom data to files */
     WriteText(&results[FIRST][0][0][0], "AB.TXT");
     WriteText(&results[SECOND][0][0][0], "CD.TXT");

     /* Write out scale factors */
     WriteScaleFactors(&scale[FIRST][0][0],     PZ*HZ*HALVES,
"SF.TXT");

     /* Acknowledge completion */
     fprintf(stderr,  "Done  generating  prom  files  and  scale
factors\n");
}

/*
 * GetInputFile - verify argument count and retrieve file name
 */
/* ---------- */
static FILE *GetInputFile()
 {
FILE *tf;

     /* Return invalid argument count */
/********************************************************************

     if (argCount != 2)
     {
          return NULL;
     }
********************************************************************/

     /* Open file  note hard coded for now */
     if ((tf = fopen("DATA.OUT", "rt")) == NULL)
     {
          perror("");
     }

     return tf;
}

/*
 * GetFSOpsRND - read scalers from input file in designed format
 */
/* ---------- */

#ifdef SPARC
#define _MAX_PATH 144      /* Note the was a MICROSOFT "C" define
within */
#endif              /* stdlib.h, SUN "C" is not supported        */
```

```
            perror("");
            exit(1);
        }
  }

  /*
   * BuildRawOutOffset - read raw outputs and offsets from input file
   */
  /* ---------- */
  static void BuildRawOutOffset(fp,off1,raw1,off2,raw2)
   FILE *fp;
   float *off1,*raw1,*off2,*raw2;
  {
  #define NUMPARMS 2
  #define LONGBUFFER 300
  #define STARTPOS  23
  static char MatchString[] = "Diag";

  char buffer[LONGBUFFER], *rc, *bufferPtr;
  float *ptr1, *ptr2;
  int hz, pz, count, bufferOffset;

      /* Find first line containing the identifying string */
      do
      {
          rc = fgets(buffer, LONGBUFFER, fp);

      }    while  ((rc  !=  NULL)  &&  strncmp(buffer,  MatchString,
                 strlen(MatchString)));

      if (rc == NULL)
      {
          perror("No data found");
          exit(1);
      }

      /* For all height zones in the file, and not at end of file
*/
      for (hz = 0; (hz < HZ) && (rc != NULL); ++hz)
      {

          /* For both sets of input data */
          for (ptr1 = off1, ptr2 = off2 + PZ - 1, count = 0;
                count < NUMPARMS;
                ptr1 = raw1, ptr2 = raw2 + PZ - 1, ++count)
          {

              /* Locate beginning of data on line */
              bufferPtr = &buffer[STARTPOS];

              /* Get the data for the left side */
              for (pz = 0; pz < PZ; ++pz, ++ptr1)
              {
                  sscanf(bufferPtr,    "    %f,%n",    ptr1,
```

```
                if    (tempStore    >    (float)full)    tempStore    =
(float)full;
                tempStore = (float)floor(tempStore /
                            (float)quantize   +   (float)0.5)
(float)quantize;
                *result++ = (unsigned char)tempStore;
            }

            /* Increment pointers */
            raw1++;
            off1++;
        }
}

/*
 * WriteIntelHex - put character data out to given file in the
intel hex
 *                         format
 */
/* ---------- */
static void WriteIntelHex(result,aLen,outFileName)
 unsigned char *result;
 int aLen;
 char *outFileName;
{
FILE *fp;
int count, tempLen, chkSum, dataCount;

    /* Open file */
    fp = fopen(outFileName, "wt");

    /* For each data group of SIZE */
    count = 0;
    while (aLen > 0)
    {

        /* Put out intel Hex line */
        tempLen = min(SIZE, aLen);
        chkSum = 0x7000 - tempLen - (count / DATABITS) - (count
% DATABITS);
        fprintf(fp, ":%02X%04X00", tempLen, count);

        /* Put out data */
        for (dataCount = 0; dataCount < tempLen; ++dataCount)
        {
            fprintf(fp, "%02X", *result);
            chkSum -= *result++;
        }

        /* Put out checksum */
        fprintf(fp, "%02X\n", chkSum % DATABITS);

        aLen -= tempLen;
        count += tempLen;
```

```
     fprintf(fp, "Channels A/B\n");

     /* Write all scale factors */
     for (count = 0; count < aLen/2; ++count)
     {
          fprintf(fp, "%2i %2i %4.1f\n", count / PZ, count % PZ,
*scale++);
     }

     fprintf(fp, "Channels C/D\n");
     for (count = 0; count < aLen/2; ++count)
     {
          fprintf(fp, "%2i %2i %4.1f\n", count / PZ, count % PZ,
*scale++);
     }

     fclose(fp);
}
```

tween the white output level and the black output level, multiplied by the quotient of said operational light intensity value less the black level threshold divided by the absolute value of the difference between the black level threshold and the white level threshold.

3. A method as recited in claim 1, wherein said black output level is zero, and said white output level is a maximum output level.

4. A method as recited in claim 1, wherein said step of storing comprises storing reference reflected light intensity information corresponding to at least black and white areas in a plurality of profiles, each one of said profiles displaying said reference reflected light intensity information plotted against one of said variables, each other of said variables being a selected constant for each profile.

5. A method for illumination compensation in a system for processing of signals representing reflected light intensities from a surface, comprising the steps of:

(a) receiving reflected light intensity information in the form of a plurality of intensity values;

(b) receiving, associated with each one of said reflected light intensity values, a value of at least one variable;

(c) remapping each of said reflected light intensity values to one of a plurality of compensated reflected light intensity values, including the steps of remapping any of said reflected light intensity values equal to or less than a preselected threshold minimum reflected light intensity value associated with the value of each of said variables for said reflected light intensity value to a minimum level output and remapping any of said reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value, associated with the value of each of said variables for each reflected light intensity value, to a maximum level output.

threshold maximum reflected light intensity value associated with the value of each of said variables for each reflected light intensity value, to a maximum level output.

9. An apparatus as recited in claim 8, further comprising memory means for remapping any of said reflected light intensity values less than said preselected threshold maximum light intensity value associated with the value of said variables for said reflected light intensity value and greater than said preselected threshold minimum reflected light intensity value corresponding to a value of each of said variables for said reflected light intensity value, linearly to a value greater than said minimum level output and less than said maximum level output.

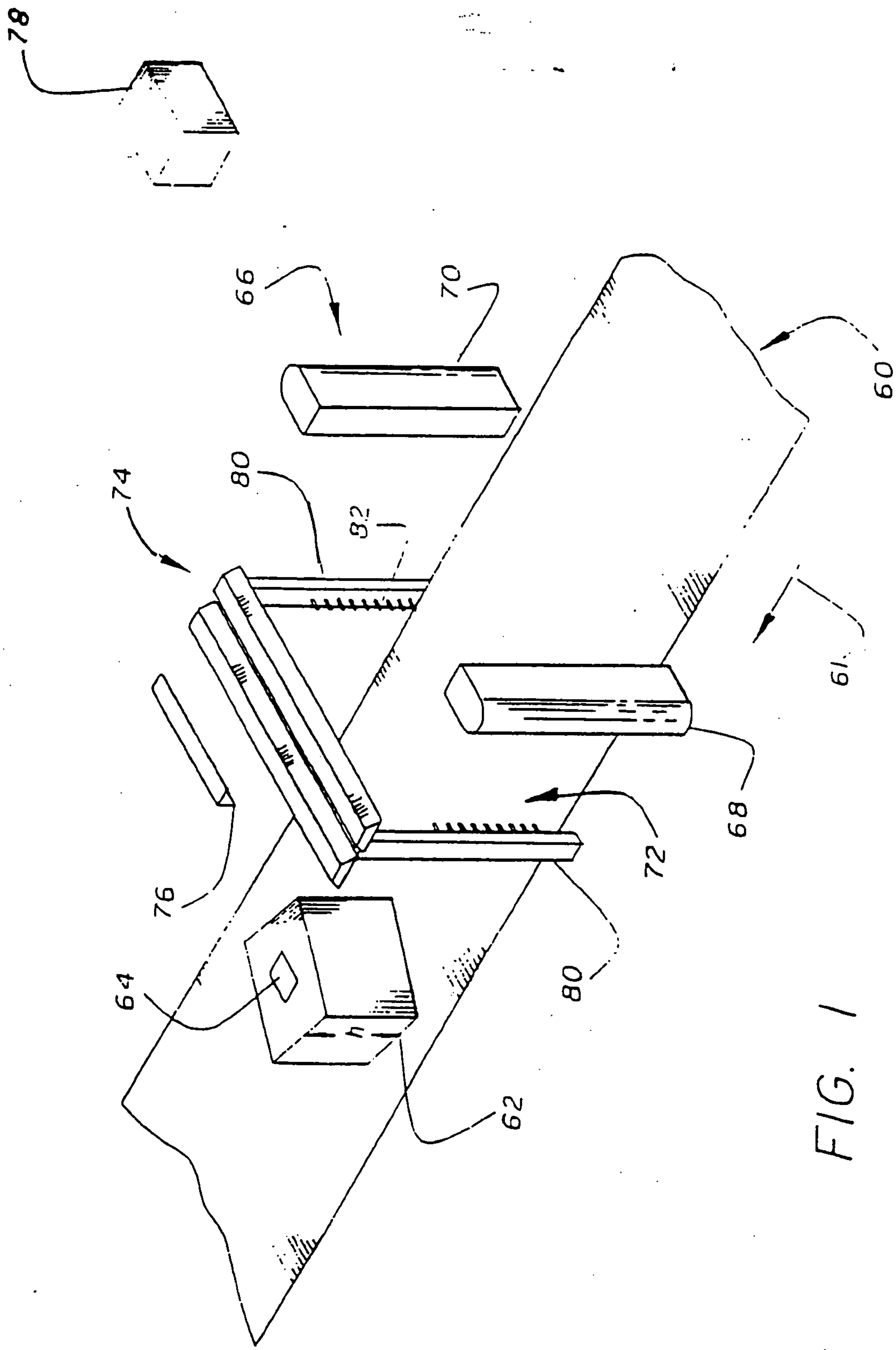


FIG. 1

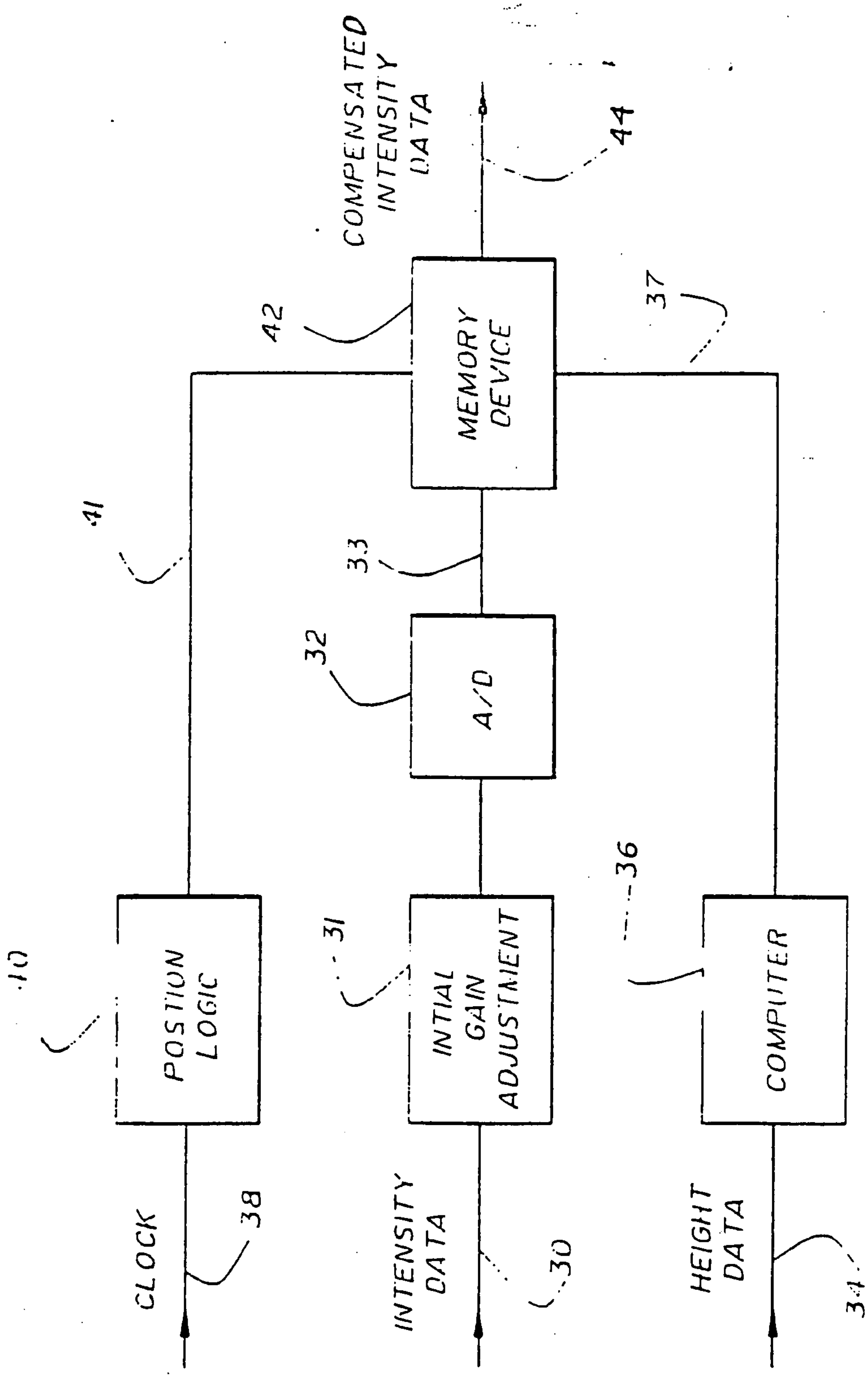

COMPENSATED INTENSITY DATA
44
MEMORY DEVICE
42
37
41
33
A/D
32
POSTION LOGIC
40
INTIAL GAIN ADJUSTMENT
31
36
COMPUTER
CLOCK
38
INTENSITY DATA
30
HEIGHT DATA
34

FIG. 2

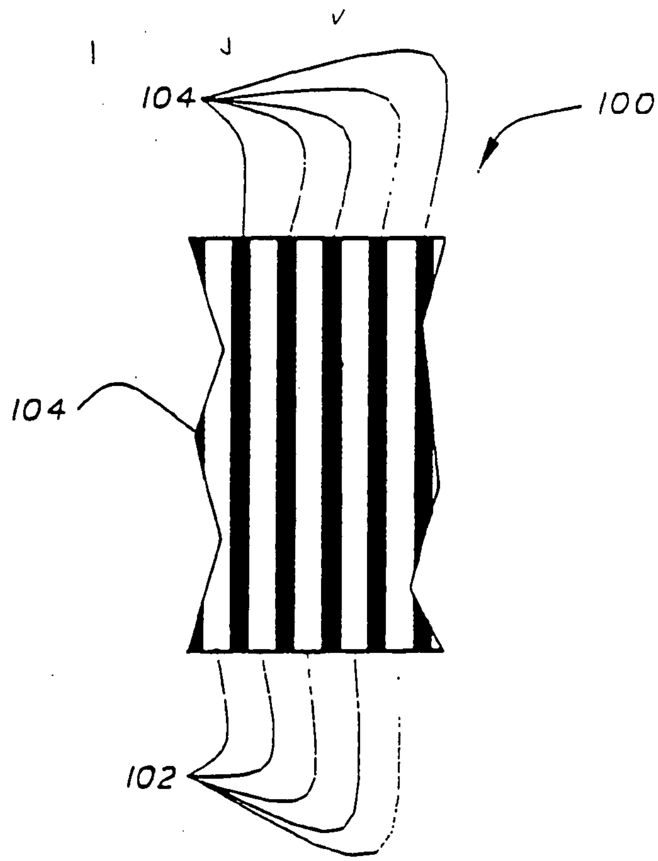


FIG. 3

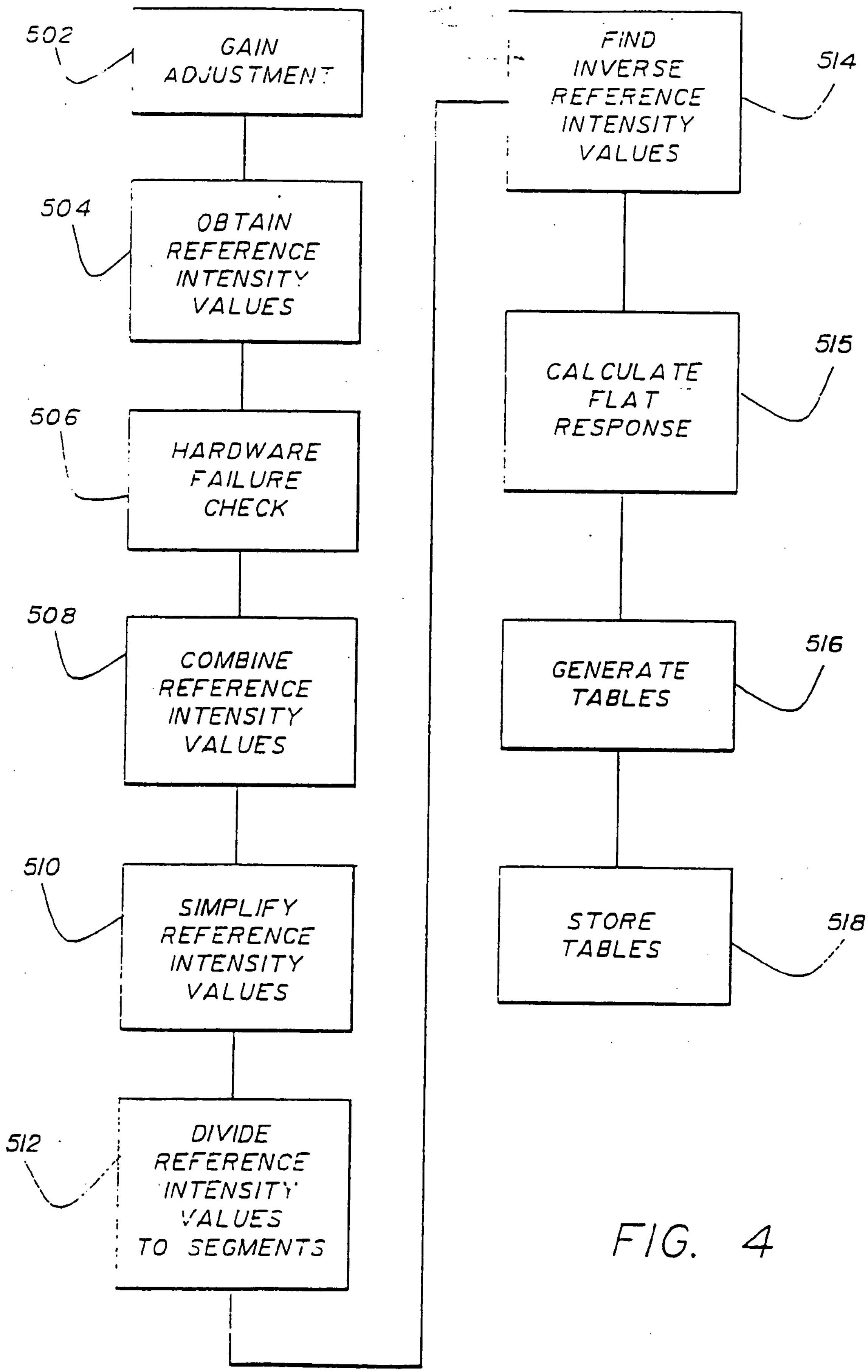

502
GAIN ADJUSTMENT
504
OBTAIN REFERENCE INTENSITY VALUES
506
HARDWARE FAILURE CHECK
508
COMBINE REFERENCE INTENSITY VALUES
510
SIMPLIFY REFERENCE INTENSITY VALUES
512
DIVIDE REFERENCE INTENSITY VALUES TO SEGMENTS
514
FIND INVERSE REFERENCE INTENSITY VALUES
515
CALCULATE FLAT RESPONSE
516
GENERATE TABLES
518
STORE TABLES

FIG. 4

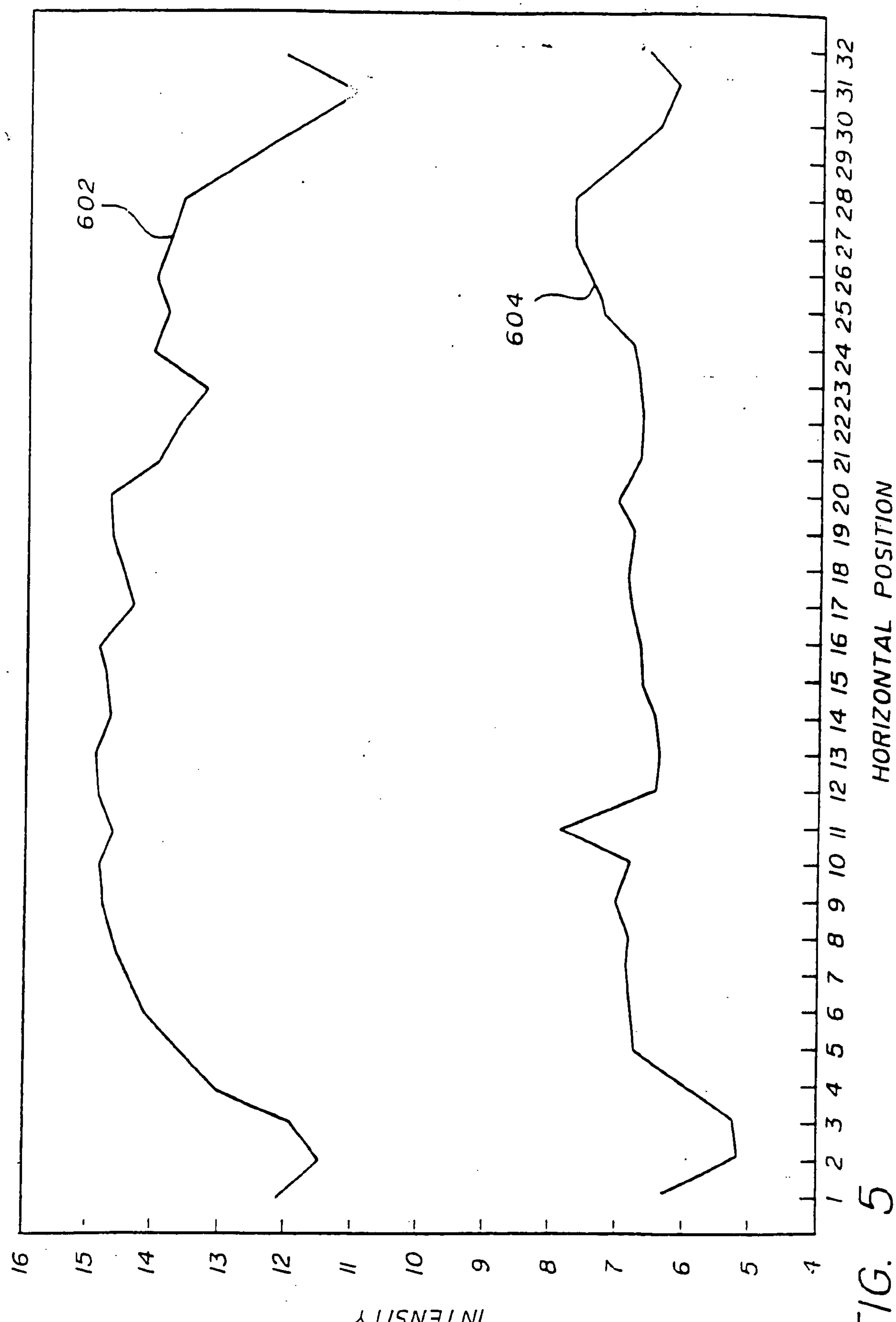

602
604
INTENSITY
16
15
14
13
12
11
10
9
8
7
6
5
4
HORIZONTAL POSITION
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

FIG. 5

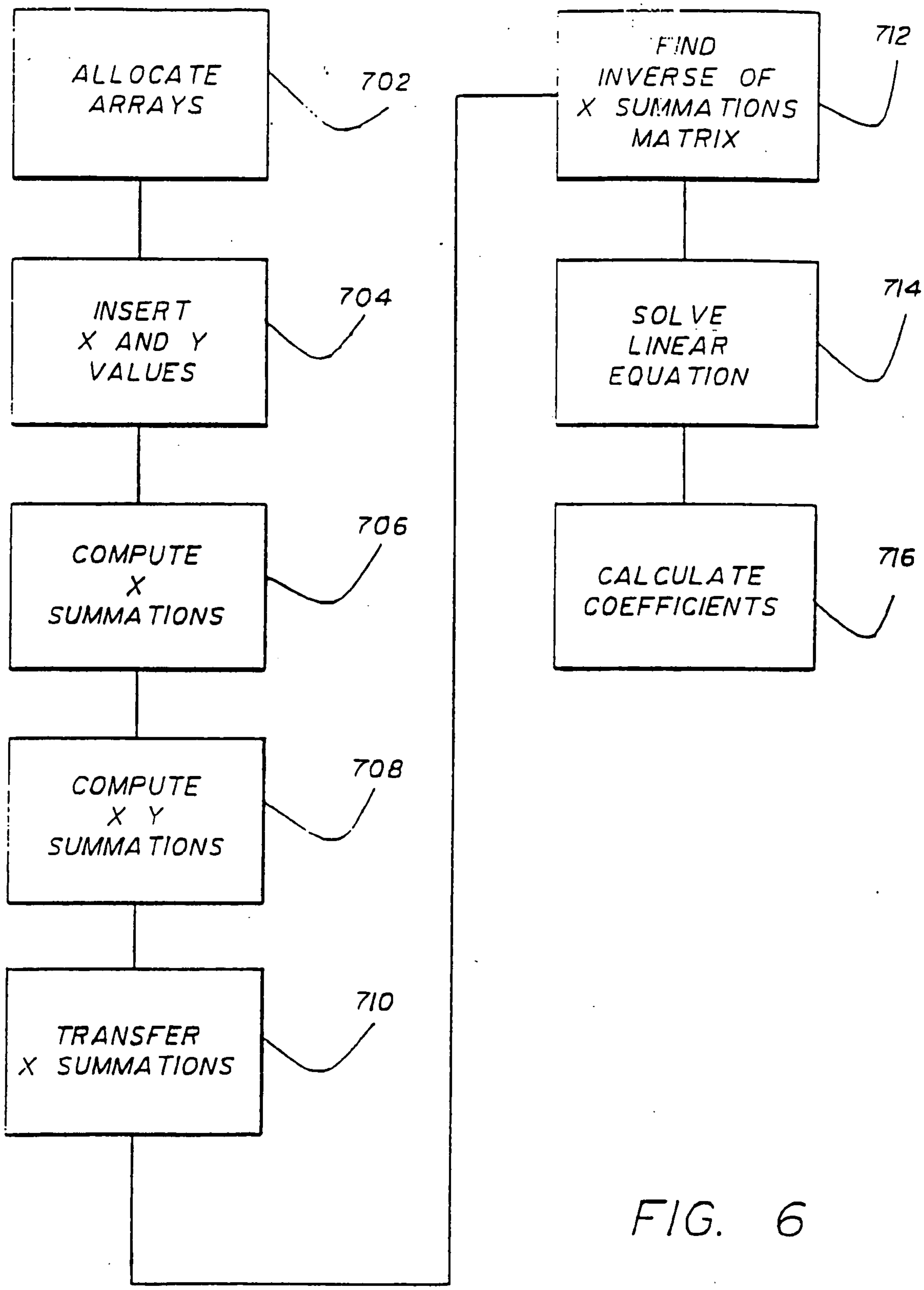

ALLOCATE ARRAYS
702
INSERT X AND Y VALUES
704
COMPUTE X SUMMATIONS
706
COMPUTE X Y SUMMATIONS
708
TRANSFER X SUMMATIONS
710
FIND INVERSE OF X SUMMATIONS MATRIX
712
SOLVE LINEAR EQUATION
714
CALCULATE COEFFICIENTS
716

FIG. 6

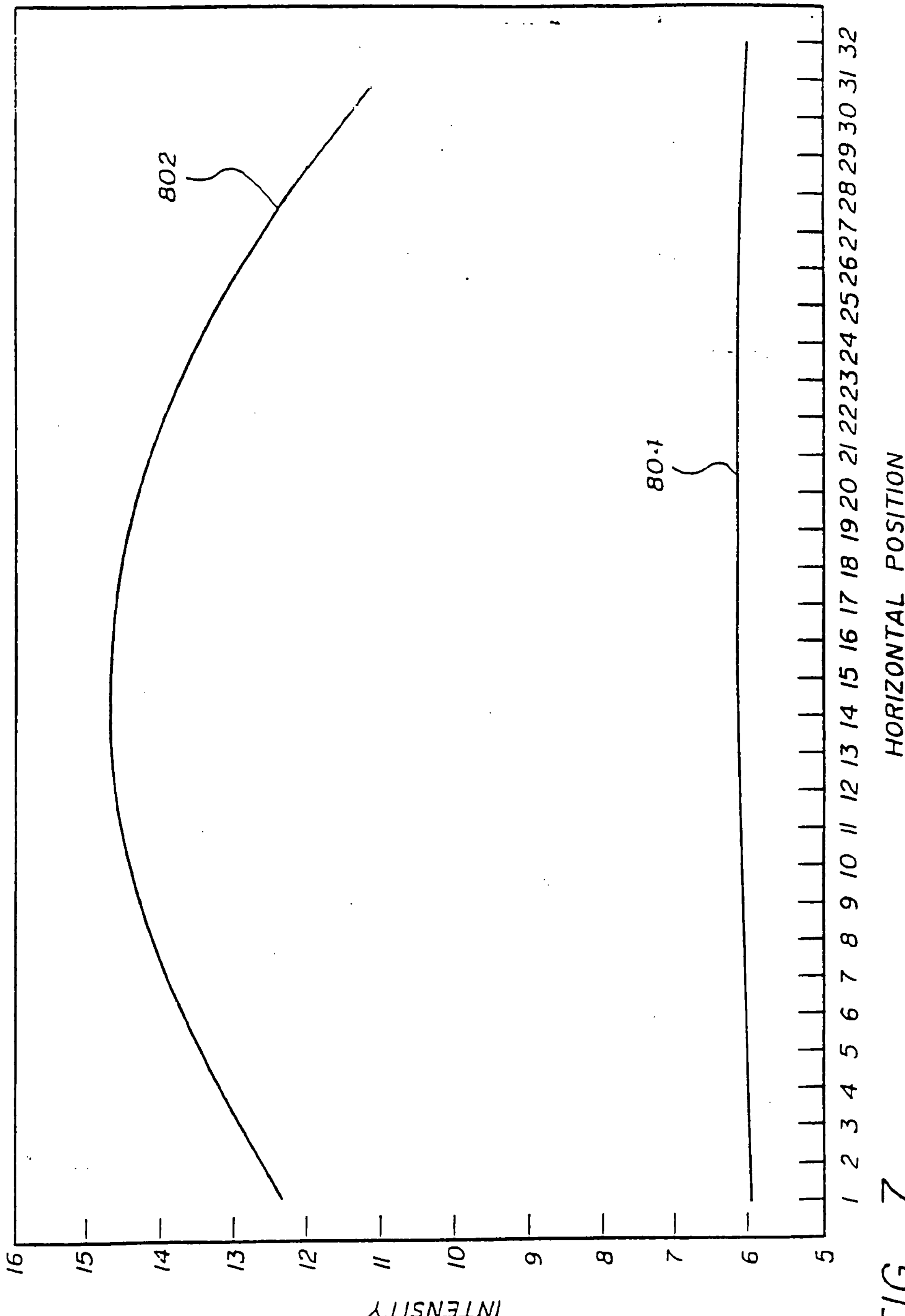

802
INTENSITY
HORIZONTAL POSITION
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
5 6 7 8 9 10 11 12 13 14 15 16

FIG. 7

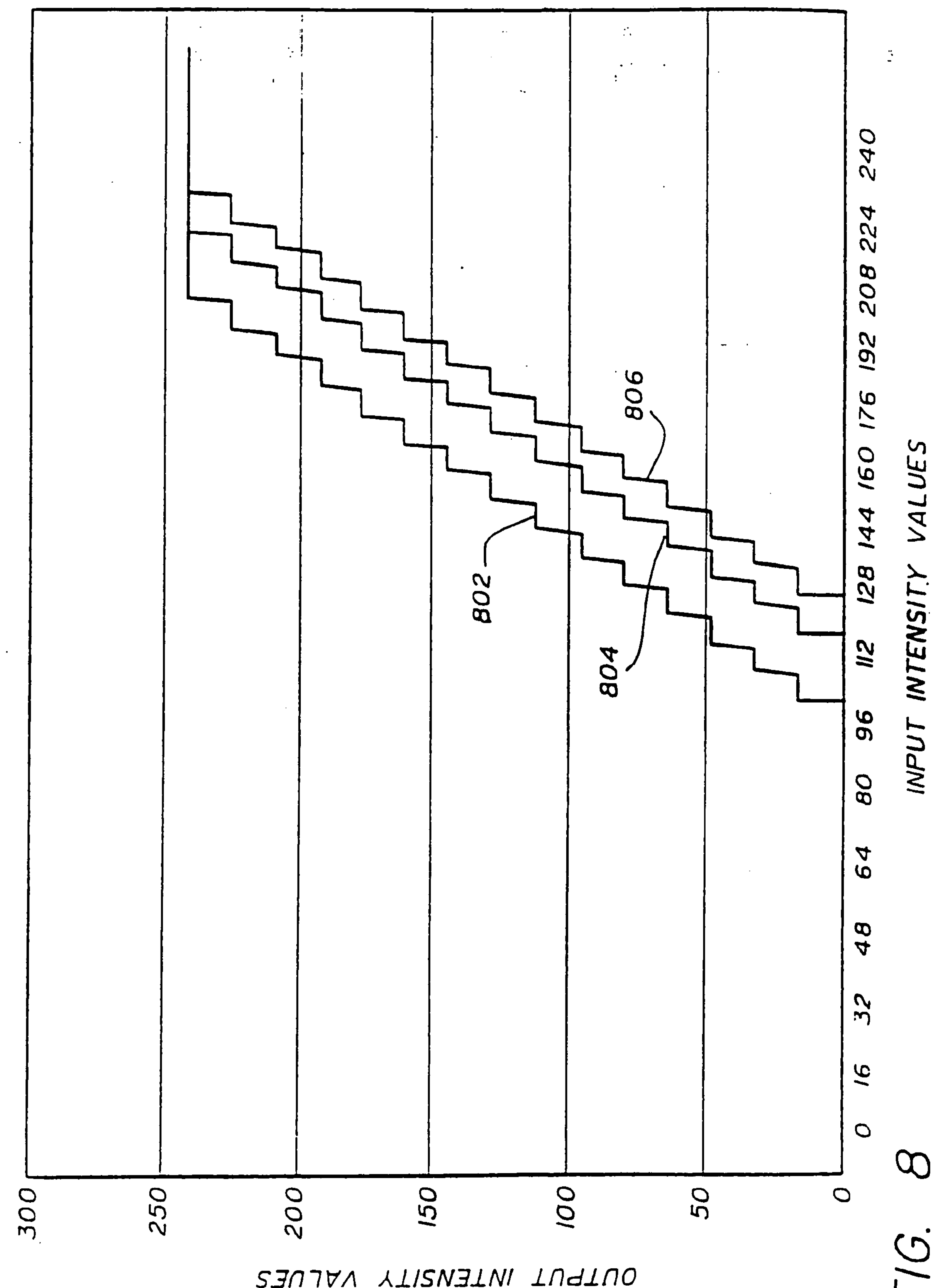

OUTPUT INTENSITY VALUES
0
50
100
150
200
250
300
INPUT INTENSITY VALUES
0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
802
804
806

FIG. 8